Neural Question Generation for Reading Comprehension

Team Name : RANberry

Ashwini Rangnekar
Nikita Agrawal
Rahil Sheth

Introduction

- Question answering has been a major area of research in Natural Language Processing (NLP) for years.
- The goal of our project is to take input paragraphs of text and output a set of relevant questions based on the information.

Baseline

Rule Based Approach

- Used modified tf-idf score to find important sentences from paragraph.
- Generated POS tags and applied NER on these sentences.
- Formulated rules which use these information to form questions.
- More than one question can be formed from each sentence.

Part-of-Speech:

Named Entity Recognition:

Results

```
Google bought IBM for 10 dollars . Mike was happy about this deal . Rahil's birthday is on 17th Dec 1995 . Rahil will pay 200
dollars to John at 7:00 PM . Nikita has 10 chocolates .

Question:  Who was happy about this deal?
Answer:  Mike was happy about this deal .

Question:  What does Nikita ha?
Answer:  Nikita has 10 chocolates .

Question:  Who will pay 200 dollars to John at 7:00 PM?
Answer:  Rahil will pay 200 dollars to John at 7:00 PM .

Question:  When willRahil pay 200 dollars to John ?
Answer:  Rahil will pay 200 dollars to John at 7:00 PM .

Question:  Who's birthday is on 17th Dec 1995?
Answer:  Rahil's birthday is on 17th Dec 1995 .

Question:  When is Rahil 's birthday ?
Answer:  Rahil's birthday is on 17th Dec 1995 .

Question:  What is Rahil?
Answer:  Rahil's birthday is on 17th Dec 1995 .
```

Limitations

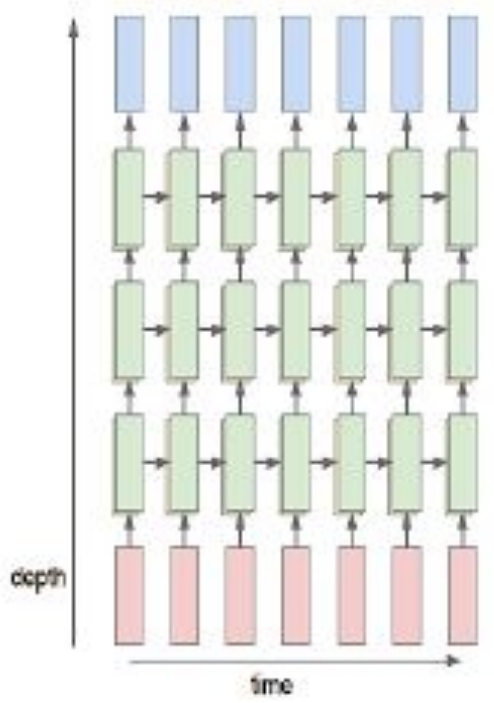
- Not a universal solution, as we need massive number of rules to catch all possibilities.
- Fails with complex and long sentences.
- Does not capture context beyond sentence.
- Questions generated are very easy to answer.
- Poor results.

Seq2Seq Model

ENCODER-DECODER

- The main network consists of two multi-layer RNNs
  - An encoder for the source language and
  - A decoder for the target language.

- We are using many-to-many RNN architecture. The reason being the input and output sentences will not have fix length.
- So we will not constraint the length of sentences.



ENCODER–DECODER

- The encoder-decoder RNN is an architecture where
  - One set of LSTMs learn to encode input sequences into a fixed-length internal representation
  - Second set of LSTMs read the internal representation and decode it into an output sequence.
- A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector.
- This makes it difficult for neural network to cope with long sentences.

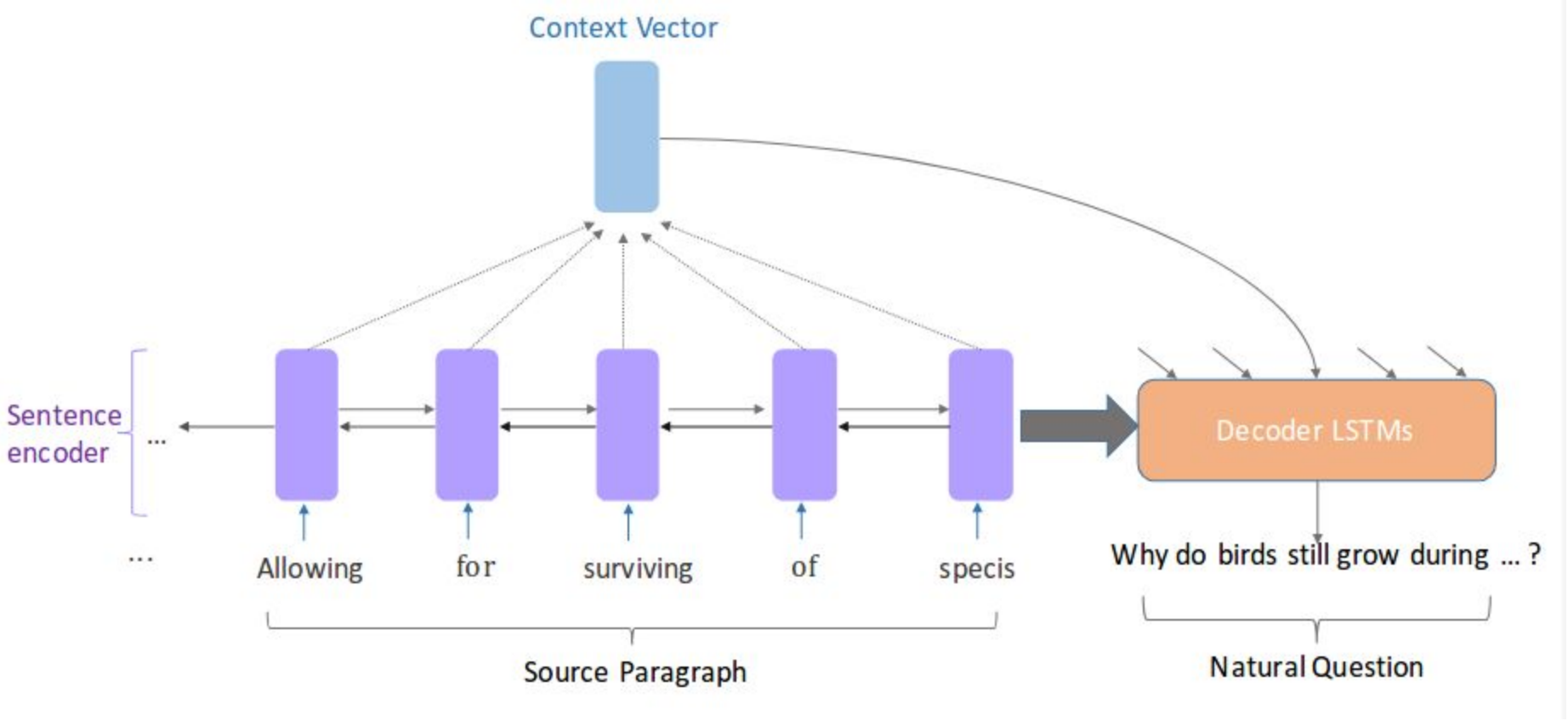
Limitations of basic Seq2Seq model

- Basic Sequence to sequence
  - All necessary information of source sentence must be compressed into fixed length vector.
  - With increase in length of sentence, fixed length vector cannot capture information of starting phrases properly and this results in worse performance for very long input sequences.

Seq2Seq with Attention Model

Attention Mechanism

- Attention is the idea of freeing the encoder-decoder architecture from the fixed-length internal representation.
- This is achieved by :
  - Keeping the intermediate outputs from the encoder LSTM from each step of the input sequence.
  - Training the model to learn to pay selective attention to these inputs and relate them to items in the output sequence.
- It encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation.



Seq2Seq model with Attention Mechanism based on Sentence encoding only

Limitations

- Fails to scale at paragraph-level let alone document-level.
- Does not capture context beyond sentence.
- Questions generated are easy to answer
- Cannot generate answer of the questions it produces.
- Results are good but not state-of-the-art.

Our Approach!

- We propose to apply neural encoder-decoder model to generate answer focused questions based on natural language sentences.
- The approach is fully data-driven with no sophisticated NLP pipelines or any hand-crafted rules/features
- The proposed approach uses a feature-rich encoder to encode answer position, POS and NER tag information.

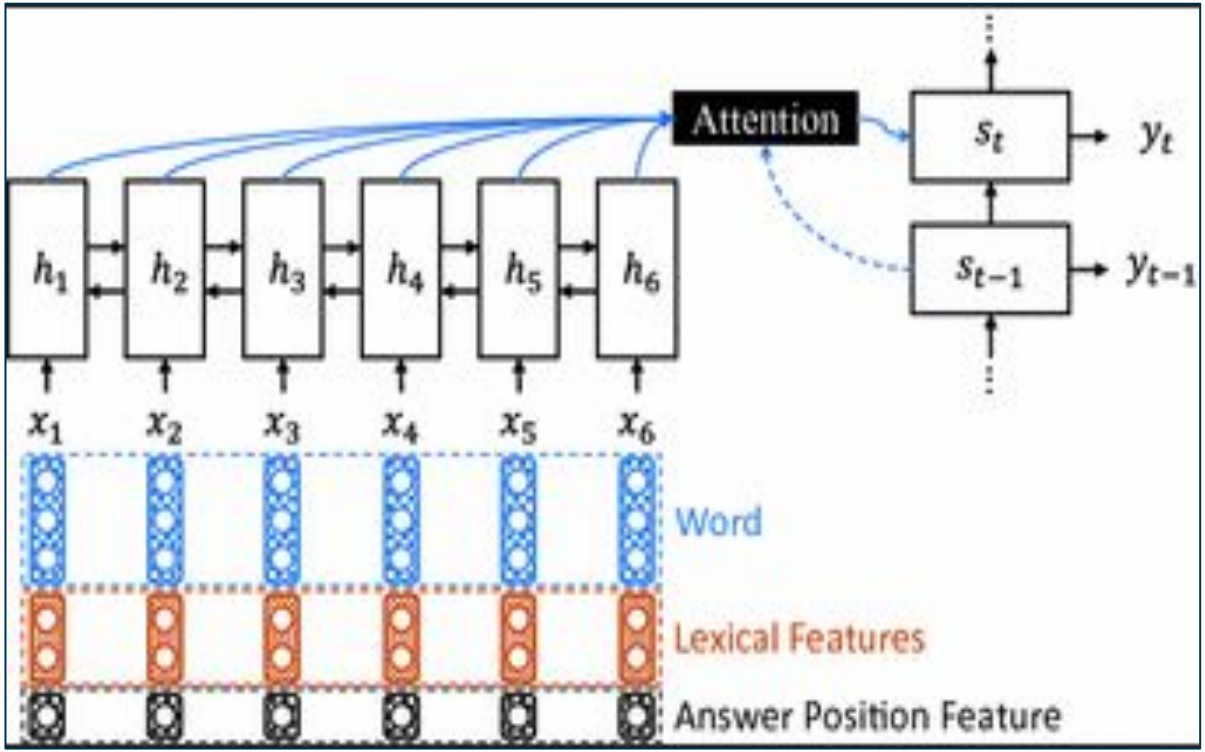
Important Sentence Selection

- First, we perform the encoding using sum operation or convolution+maximum pooling operation over the word vectors comprising each sentence in the input paragraph.

- Then we use a bidirectional LSTM to encode the paragraph for sentence-level sequence labeling. The input is a paragraph consisting of sentences, whose encoded representation is fed into each hidden unit.

Enriching Encoder

- We use Gated Recurrent Unit (GRU) To capture more context information, we use bidirectional GRU (BiGRU) to read the inputs in both forward and backward orders.
- BiGRU encoder not only reads the sentence words, but also handcrafted features, to produce a sequence of word-and-feature vectors.
- We concatenate the word vector, lexical feature embedding vectors and answer position indicator embedding vector as the input of BiGRU encoder.



$$s_t = \text{GRU}(w_{t-1}, c_{t-1}, s_{t-1})$$

$$s_0 = \tanh(\mathbf{W}_d \tilde{h}_1 + b)$$

$$e_{t,i} = v_a^\top \tanh(\mathbf{W}_a s_{t-1} + \mathbf{U}_a h_i)$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i=1}^{n} \exp(e_{t,i})}$$

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_i$$

Overview of the Neural Question Generation (NQG) framework

Model	Dev set	Test set
Seq2seq + att	3.01	3.06
NQG	10.06	10.13
NQG+	12.3	12.18
NQG++	13.27	13.29

BLEU evaluation scores

References

- *Attention-over-Attention Neural Networks for Reading Comprehension*
- *Learning to Ask: Neural Question Generation for Reading Comprehension*
- *Identifying Where to Focus in Reading Comprehension for Neural Question Generation*
- *Simple and Effective Multi-Paragraph Reading Comprehension*
- *Neural Question Generation from Text: A Preliminary Study*
- Stanford CoreNLP

Thank You!